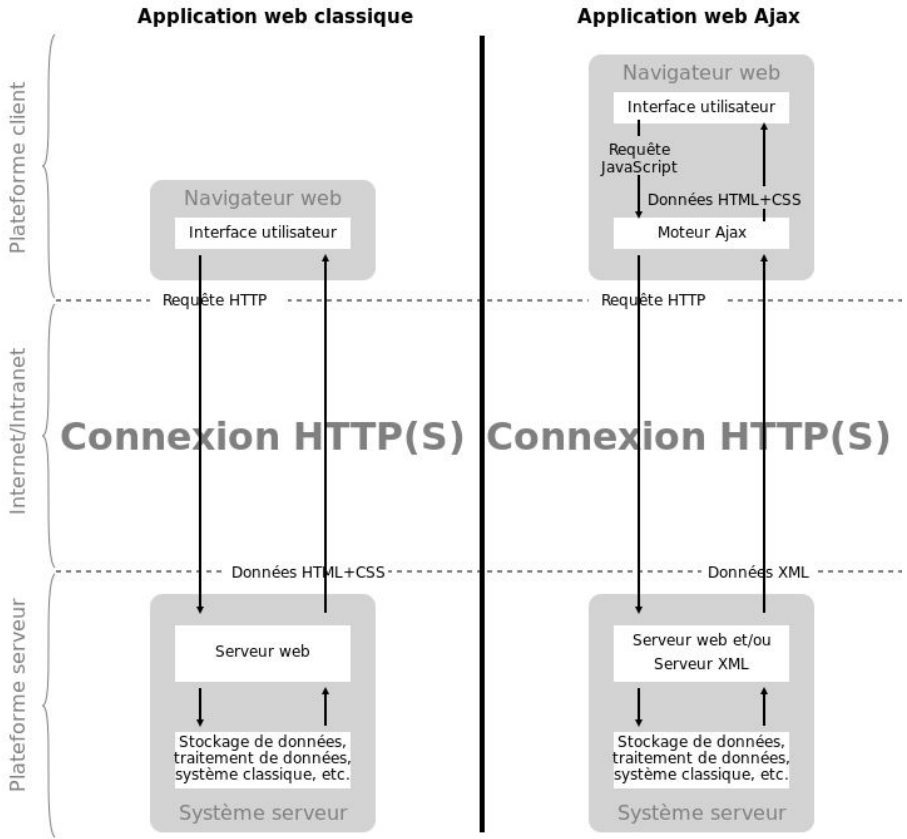
Javascript

Session 3: Ajax et jQuery

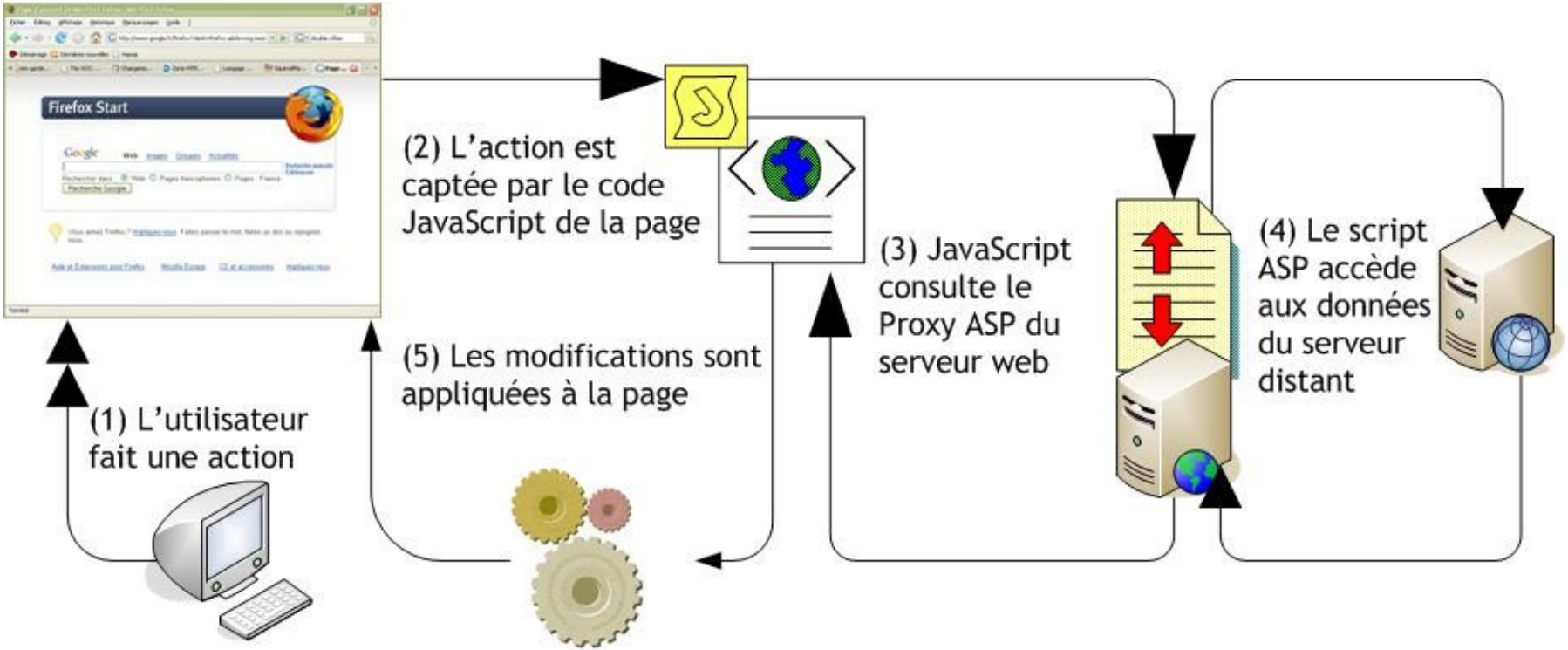
Thibault Clérice
@ponteineptique (Twitter et Github)
contact@algorythme.net

AJAX
Asynchronous Javascript And XML

- Communication asynchrone entre deux services.



Ajax : exemple

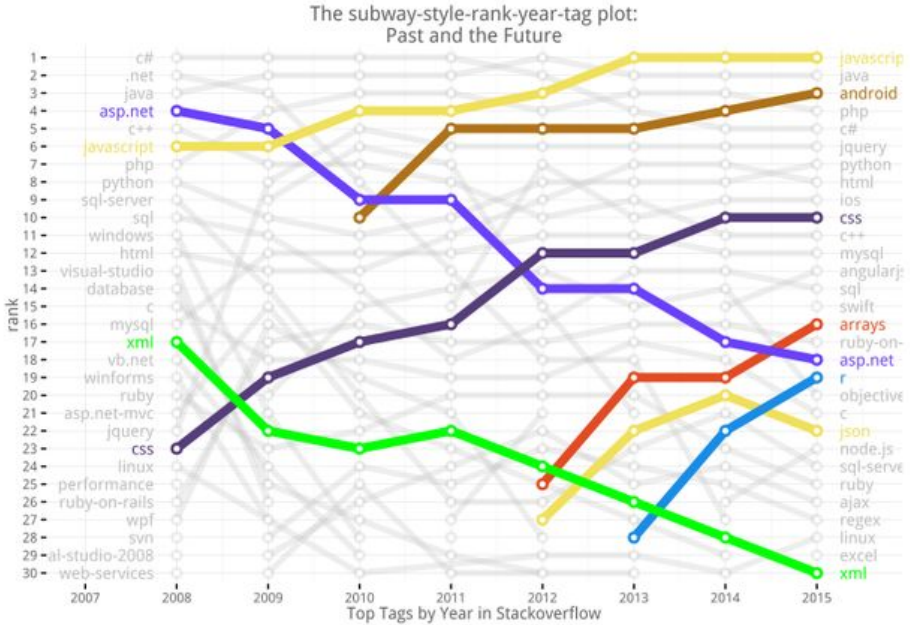


HTTP : les types de requetes

- GET : obtenir une information
- POST : Créer ou envoyer une information
- PUT : Créer ou mettre à jour une information
- DELETE : Supprimer une ressource
- OPTIONS : Savoir quelles sont les verbes disponibles

JSON

- JavaScript Object Notation
- JSON-LD : JSON for Linking Data
- Format simple, lisible, rapide
- Majoritaire dans le monde du privé



Structure : JSON vs XML

	XML	JSON
Caractères de contrôle	<>=”’/:\s	,”{}[]’:
Données complexes, uniques	Meilleur	
Données “simples”, répétitives		Meilleur
Rapidité		Meilleur
Disponibilité dans les langages		Meilleur
Légèreté		Plutôt
Communication Machine	?	?

Structure : JSON (Object Notation) vs XML (Markup)

```
<Document>
   <Paragraph Align="Center">
      Here <Bold>is</Bold> some text.
   </Paragraph>
</Document>
```

```
{
   "Paragraphs": [
      {
         "align": "center",
         "content": [
            "Here ", {
               "style" : "bold",
               "content": [ "is" ]
            },
            " some text."
         ]
      }
   ]
}
```

Structure : JSON (Object Notation) vs XML (Markup)

```
<Person>
   <FirstName>Homer</FirstName>
   <LastName>Simpsons</LastName>
   <Relatives>
      <Relative>Grandpa</Relative>
      <Relative>Marge</Relative>
      <Relative>The Boy</Relative>
      <Relative>Lisa</Relative>
      <Relative>I think that's all of
them</Relative>
   </Relatives>
</Person>
```

```
{
   "firstName": "Homer",
   "lastName": "Simpson",
   "relatives": [ "Grandpa", "Marge", "The
Boy", "Lisa", "I think that's all of them" ]
}
```

AJAX en Javascript

Problème :

- Long à écrire
- Pas toujours bien implémenté

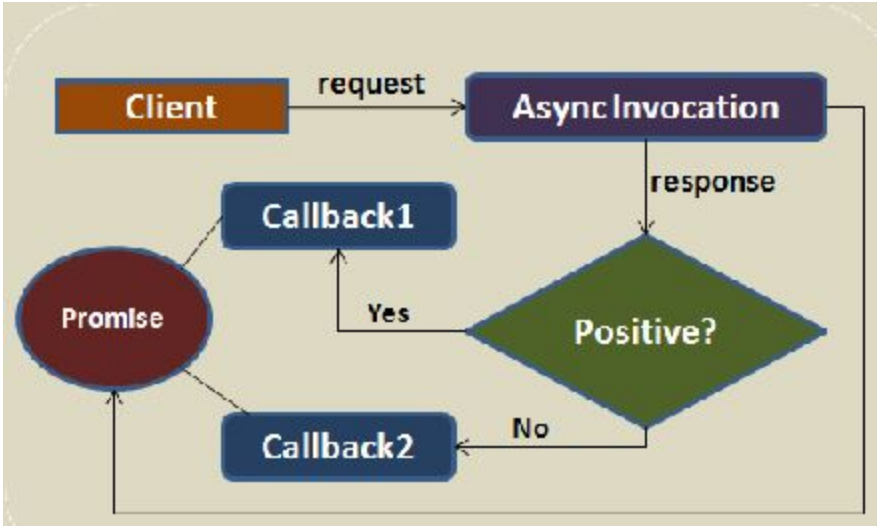
https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_callback

```
<!DOCTYPE html>
<html>
<body>
<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button"
onclick="loadDoc('ajax_info.txt', myFunction)">Change Content
</button>
</div>
<script>
function loadDoc(url, cFunction) {
  var xhttp;
  xhttp=new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      cFunction(this);
    }
  };
  xhttp.open("GET", url, true);
  xhttp.send();
}
function myFunction(xhttp) {
  document.getElementById("demo").innerHTML =
  xhttp.responseText;
}
</script>
</body>
</html>
```

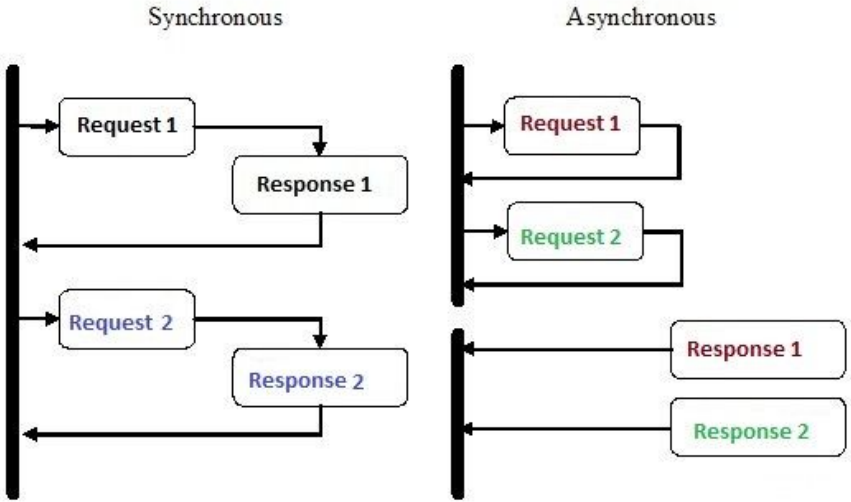
La notion de callback (1)

```
var x = function() {
    console.log("Ecole Nationale")
};

var y = function(callback) {
    callback();
    console.log("Des Chartes");
};

y(x);
```

La notion de callback (2)

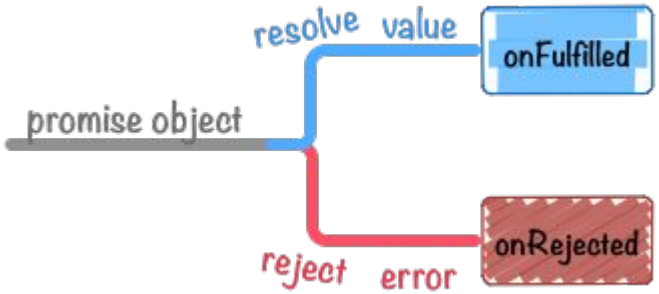


La notion de callback (3)




```
<!DOCTYPE html>
<html>
<body>
<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button"
onclick="loadDoc('ajax_info.txt', myFunction)">Change Content
</button>
</div>
<script>
function loadDoc(url, cFunction) {
  var xhttp;
  xhttp=new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      cFunction(this);
    }
  };
  xhttp.open("GET", url, true);
  xhttp.send();
}
function myFunction(xhttp) {
  document.getElementById("demo").innerHTML =
  xhttp.responseText;
}
</script>
</body>
</html>
```

Ajax, Callback : Les promesses




```
var p = new Promise(
  function(resolve, reject){
  ...
  if(something)
     resolve({});
  else{
     reject(new Error());
  }
})
```

```
p.then(
function(data){
  ...
},
function(err){
  ...
}
);
```

Être développeur-se, c’est être fainéant-e

jQuery

- Depuis 2006
- Librairie / Bibliothèque
- Ensemble de fonctionnalités simplifiées
- Plus grands gains (au moins à l'époque) :
  - Sélecteur CSS avec la fonction $(selecteur)
  - Ajax (http://api.jquery.com/jquery.ajax/)
- Cross-browser :
  - Pas de problème (ou moindre) du type Internet Explorer

```
$("#id > nomdetag.classe:first");

$.ajax();
$.get();
// Version promesse
var jqxhr = $.ajax( "example.php" )
  .done(function() {
    alert( "success" );
  })
  .fail(function() {
    alert( "error" );
  })
  .always(function() {
    alert( "complete" );
  });
jqxhr.always(function() {
  alert( "second complete" );
});
```

Bibliothèque/Librairie, Framework, Plugin, Script

Propose un ensemble de fonctionnalités pour appeler votre coder ou être appelé par votre code

Réutilise des frameworks

React
AngularJS
jQuery

jQueryUI

Améliore un framework ou une librairie

Votre Code

Utiliser jQuery (et autres utilitaires externes)

Général

- Avant votre code
- Mieux à la fin de votre page (avant </body>)
- Dans des balises script

```
<script type="text/javascript" src="URL"></script>
```

jQuery (Supplémentaire)

Mettre votre code d'exécution comme suit :

```
<script type="text/javascript">
$(document).ready(function($){
    // Votre code ici !
});
</script>
```

Les fonctions et sélecteurs jQuery

https://oscarotero.com/jquery/

SELECTORS

Basics
*
.class
element
#id
selector1, selectorN, ...

Hierarchy
parent > child
ancestor descendant
prev + next
prev ~ siblings

Basic Filters
:animated
:eq()
:even
:first
:gt()
:header
:lang()
:last
:lt()
:not()
:odd
:root
:target

Content Filters
:contains()
:empty
:has()
:parent

Visibility Filters
:hidden
:visible

Attribute
[name|="value"]
[name*="value"]
[name~="value"]
[name$="value"]
[name="value"]
[name!="value"]
[name^="value"]
[name]
[name="value"][name2="value2"]

Child Filters
:first-child
:first-of-type
:last-child
:last-of-type
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type()

Forms
:button
:checkbox
:checked
:disabled
:enabled
:focus
:file
:image
:input
:password
:radio
:reset
:selected
:submit
:text

Les événements jQuery

- Lier une action à un événement
- Un événement est une action liée à la page web :
  - Click
  - Hover
  - Dblclick

Lors de l'appel d'un événement, l'élément au départ du quel il a été déclenché appelle la fonction anonyme ici créée. Cet élément est un objet et est représenté en javascript par *this*.

```
<script type="text/javascript">
$(document).ready(function($){
    $("a").on("click", function(evenement) {
        evenement.preventDefault();
        var element = $(this);
        console.log("Texte : " + element.text() + " Adresse : "
+element.attr("href"));
    });
});
</script>
```

Ecrire des fonctions : exercice

A l'aide de la fonction trouverlemot précédemment créée, créer une interface permettant de chercher dans les fichiers json du slide 20 et l'interface suivante.

Commencer avec Remedy of love et Art of beauty

Penser à effacer les résultats avant chaque nouvelle recherche !

Vous aurez besoin de :

?